Práctica 2

Ejercicios

1

¿Qué sucedería si la tarea task2 (más prioritaria) ejecutase una espera activa en un bucle hasta que se cumpliera el plazo de tiempo para una siguiente iteración? ¿La tarea task1 podría ejecutarse?

La tarea menos prioritaria (tarea 2) nunca podría llegar a ejecutarse, ya que siempre lo estaría haciendo la tarea más prioritaria (tarea 1).

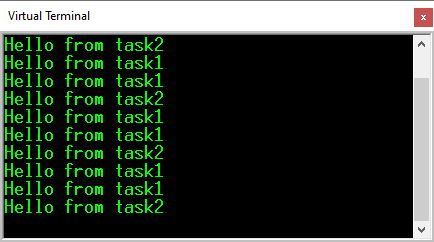
\*TAREA IDLE: Cuando ninguna de las tareas se esta ejecutando, se pasa a ejecutar está tarea que esta en segundo plano y se encarga de mantener la actividad en la CPU.

2

Cree una nueva instancia de la tarea task1 con el mismo tamaño de pila y prioridad 3 mediante xTaskCreate().

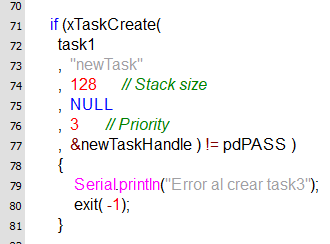
Al tener la tarea 1, la misma prioridad que la nueva tarea creada (tarea 3), el sistema FreeRTOS automáticamente otorga más prioridad a una tarea y en el siguiente instante de ejecución en el que ambas tareas vuelvan a coincidir, otorgará más prioridad a la otra.

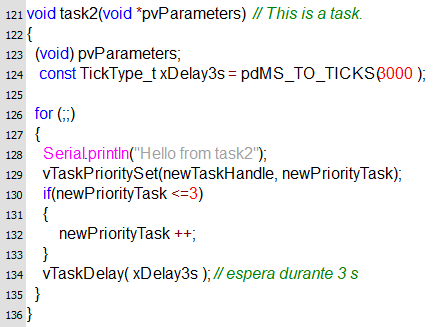
Al ejecutar las mismas líneas de código, el mensaje que muestra por el terminal será idéntico y no habrá posibilidad de observar esta alternancia de prioridad entre ambas tareas.



vTaskPrioritySet()

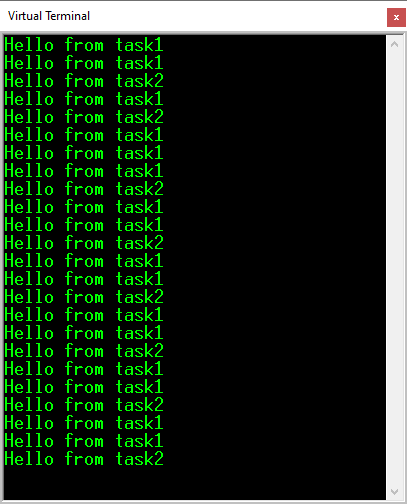






Con el código anterior, la nueva tarea creada se define la misma prioridad que la tarea 1 y referencia al mismo fragmento de código. La primera vez que se ejecuta la tarea 2, indica que la prioridad de la nueva tarea es 0. Después, cada ejecución de la tarea dos incrementa la prioridad de la nueva tarea hasta alcanzar la prioridad 3.

\*NOTA: Recordar que la tarea 1 tiene una prioridad de 3 y la tarea 2 de 1.



Tarea nueva -> Prioridad 3

Tarea nueva -> Prioridad 2

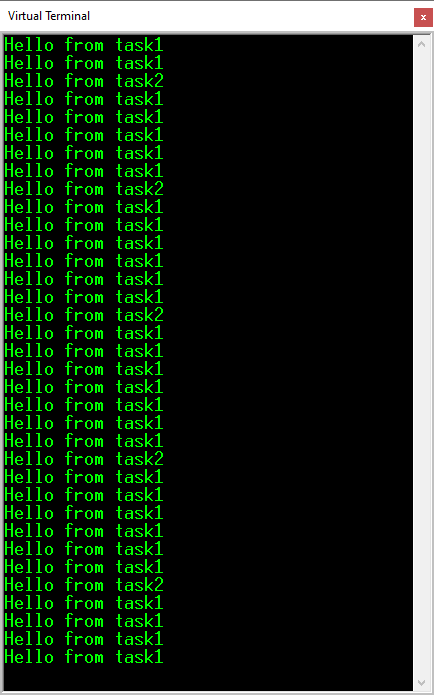
Tarea nueva -> Prioridad 1

Tarea nueva -> Prioridad 0

Tarea nueva -> Prioridad 3

3

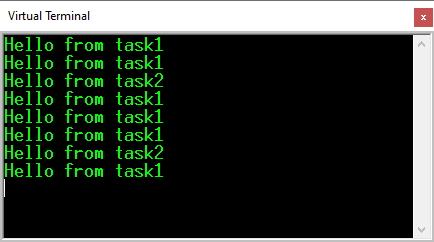
Modifique el tiempo de suspensión de la tarea task1, introduciendo el valor de 1 s en lugar de 3 s. Guarde el fichero, reconstruya el proyecto y ejecútelo. Observe el resultado sobre el terminal virtual.



La tarea 1 y la nueva se ejecutan al principio por tener mayor prioridad. Después se repiten 3 veces cada una, antes de que la tarea 2 vuelva a ejecutarse.

\*NOTA: Se ha quitado la anterior modificación de la prioridad de una tarea, ahora la tarea 1 y la nueva, mantienen la prioridad 3 constantemente.

vTaskSuspend()



En esta imagen, se inicia la tarea nueva y la tarea 1 con 1 segundo de espera activa, mientras que la tarea 2 con 3 segundos. Esta tarea dos ahora incluye una nueva línea en su código que suspenderá la tarea nueva, por esta razón, tras la ejecución de la tarea 2 solamente se ven 3 ejecuciones de la tarea 1.

4